



徐爱钧 彭秀华 编著

单片机高级语言 **C51** 应用程序设计

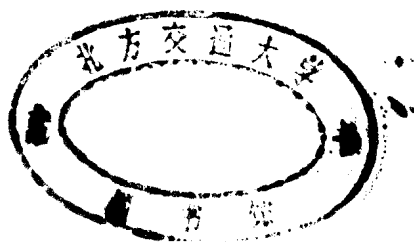
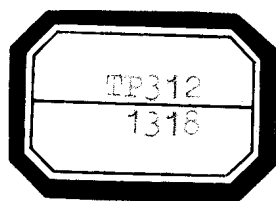


电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.com.cn>

单片机高级语言 C51 应用程序设计

徐爱钧 彭秀华 编著



電子工業出版社

Publishing House of Electronics Industry

内 容 简 介

本书以美国 Franklin 软件公司专为 8051 系列单片机设计的高效率 C 语言编译器为核心,阐述了单片机高级语言 C51 的基本语法、各种运算符的作用以及语句结构等,详细介绍了 C51 编译器、L51 连接定位器以及 A51 宏汇编器的使用方法,对于各种编译连接控制指令作了详尽的描述,并给出了各种编译连接错误信息。另外还详细介绍了 Franklin 公司独具特色的软件仿真器 DS51 的使用方法。采用 DS51 可在完全没有 8051 单片机硬件的条件下,用一台普通 PC 机即可完成 8051 应用系统程序的仿真调试工作,这一点对于采用高级语言编程的单片机用户是十分有益的。本书的特点是强调实用性和先进性,全书各章中都列举了大量的程序例子,并且还有一章(第十章)专门用来阐述 C51 应用编程技巧和实例,列举了一些 C51 应用中常见的问题与解决方法,给出了若干采用 C51 高级语言进行单片机应用程序设计的实例。

本书适合于广大从事单片机应用系统开发研制的工程技术人员阅读,特别适合于已经了解单片机原理而希望采用高级语言编程的单片机用户,也可作为普通高等工科院校大学生或研究生学习 C 语言的教学用书。

图书在版编目(CIP)数据

单片机高级语言 C51 应用程序设计/徐爱钧,彭秀华编
著,一北京:电子工业出版社,1998.4
ISBN 7-5053-4537-0

I. 单… II. ①徐… ②彭… III. C 语言-程序设计 IV.
TP312

中国版本图书馆 CIP 数据核字(98)第 06181 号

书 名:单片机高级语言 C51 应用程序设计

编 著:徐爱钧 彭秀华

责任编辑:李新社

特约编辑:康宗朗

排版制作:电子工业出版社计算机排版室

印 刷 者:北京天宇星印刷厂

装 订 者:河北涿州桃园装订厂

出版发行:电子工业出版社出版、发行 URL: <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话:68214070

经 销:各地新华书店

开 本:787×1092 1/16 印张:28.75 字数:736 千字

版 次:1998 年 6 月第 1 版 1998 年 6 月第 1 次印刷

书 号:ISBN 7-5053-4537-0
TP·2128

定 价:45.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换。

版权所有·翻印必究

前 言

随着计算机应用技术的不断发展,在工业测量控制领域内单片机的应用越来越广泛。同时,随着超大规模集成电路工艺和集成制造技术的不断完善,单片机的硬件集成度也在不断提高,已经出现了能满足各种不同需求、具有各种特殊功能的单片机。就 8051 系列单片机而言,由于 Intel 公司将 8051 CPU 内核向全世界各大半导体公司的扩散,目前已有 Philips、Siemens、Dallas、OKI、Advance Micro Device、Atmel 等公司生产 100 多种型号的 80C51 系列单片机。这类单片机具有集成度高,性能价格比优越的特点,在工业测量控制领域内获得了极为广泛的应用,在今后的相当一个时期内,80C51 单片机仍将是我国的主流机种。在开发一个单片机应用系统时,系统程序的编写效率在很大程度上决定了目标系统的研制成效。早期在研制单片机应用系统时,大多以汇编语言作为软件工具。汇编语言程序能够直接操作机器硬件,指令的执行速度快。但由于汇编语言不是一种结构化的程序设计语言,汇编语言程序较难编写和调试,程序本身的编写效率较低。随着单片机硬件性能的提高,其工作速度越来越快,目前 80C51 单片机的最高时钟频率可达 40MHz 以上。因此在编写单片机应用系统程序时,更着重于程序本身的编写效率。为了适应这种要求,现在的单片机开发系统,除了配备有汇编语言软件之外,很多还配备了高级语言软件,如 C51、PLM51 等。

C 语言是一种通用的计算机程序设计语言,它既有高级语言的各种特征,又能直接操作系统硬件,而且可以进行结构化程序设计,用 C 语言编写的程序很容易移植。近年来已出现了若干种专为微型计算机设计的 C 语言编译器,如美国 Franklin 软件公司推出的 Franklin C51 就是一种专为 8051 系列单片机设计的高性能的 C 编译器,它采用符合 ANSI 标准的 C 语言编程,能够产生极高速度和及其简洁形式的目标代码,在代码效率和执行速度上完全可以和汇编语言相比,并且具有十分丰富的库函数可供用户直接调用,从而极大地提高了程序的编写效率。C51 编译器支持模块化程序设计,各模块源程序用符合 ANSI 标准的 C 语言编写,然后由 C51 编译器生成一个可重新定位的目标文件。简单的应用程序只要一两个模块就能满足要求,复杂的应用程序可能会由 100 多个模块所组成。各个模块在编译正确之后,由连接定位器 L51 进行连接定位,产生可执行的绝对目标文件。L51 可以自动从 LIB51 库中选出所需要的库函数与各目标模块连接在一起,组成一个完整的应用程序。Franklin 公司还提供一种适用于 IBM-PC 机的仿真调试软件 DS51,它可以在完全没有 8051 单片机硬件的条件下,完成 C51 应用程序的仿真和调试工作。由此可见采用 C 语言编程来解决单片机系统的程序设计问题要比采用汇编语言编程容易得多和有效得多。目前新推出的单片机开发系统大都配置了 C51 高级语言,广大单片机用户及工业应用系统的开发研制人员都迫切希望了解 C51 高级语言的特性,掌握用 C51 编写 8051 单片机应用程序的技术和方法。本书就是为了满足这一需要,结合我们自己应用 C51 编程的经验编写而成的。

本书以 Franklin C51 为基础,阐述 C 语言的一些基本特征和应用编程技巧。为使从未接触过 C 语言的读者能够较快地掌握用 C51 编写单片机应用程序的方法,书中通过大量程序实例来描述 C 语言的基本语法、各种运算符的作用、数据类型和变量的存储结构、函数的调用等,

以帮助读者加深对于采用 C51 编写单片机应用系统程序方法的理解。C51 是 C 语言的一种特殊扩充,因此在阐述的过程中对 C51 本身所独有的特点着重进行了介绍,旨在帮助读者尽快掌握采用 C51 进行应用程序设计的方法和技巧。全书共分十章。第一章介绍 C 语言的一般知识以及 C51 程序的编译连接和调试的基本方法。第二章至第六章阐述采用 C51 进行程序设计的基本语法,介绍了各种数据类型、运算符、表达式及语句、函数、数组和指针、结构和联合以及预处理器命令等。第七章介绍 C51 编译器,详细介绍了各种编译控制指令、C51 的数据调用协议、与汇编语言程序的接口方法、给出了 C51 编译器的全部库函数和错误信息。第八章介绍目标文件的连接和转换,详细介绍了连接定位器 L51 的各种连接控制指令、符号转换程序 OHS51、库管理程序 LIB51 以及宏汇编器 A51 的使用方法。第九章详细介绍了 DS51 软件仿真器的使用方法。第十章介绍 C51 应用编程技巧和程序实例,阐述了 C51 实际应用中的一些常见问题和解决方法,给出了若干个采用 C51 进行实际单片机应用系统程序设计的例子。本书各章中所列举的程序例子都在 IBM-PC 机上经 DS51 调试通过。

本书在编写过程中得到朱清祥、丁福坤、燕满、张玉芳、郭邦云、王珍、邓骊等同志的协助,北京航空航天大学电子工程系康宗朗教授对本书的出版给予了热情支持并做了大量有益的工作,在此一并表示衷心的感谢。

由于作者水平有限,书中难免会有错误和不妥之处,恳请广大读者批评指正。

编著者

目 录

第一章 C 语言的基本知识	(1)
§ 1.1 C 语言的特点及其程序结构	(1)
§ 1.2 C 语言的标识符和关键字	(5)
§ 1.3 单片机 C51 程序的编译连接与调试	(7)
第二章 C51 程序设计的基本语法	(17)
§ 2.1 数据类型	(17)
§ 2.2 常量	(19)
§ 2.3 变量及其存储模式	(21)
§ 2.4 用 typedef 重新定义数据类型	(24)
§ 2.5 运算符与表达式	(25)
2.5.1 赋值运算符	(25)
2.5.2 算术运算符	(26)
2.5.3 增量和减量运算符	(26)
2.5.4 关系运算符	(29)
2.5.5 逻辑运算符	(31)
2.5.6 位运算符	(32)
2.5.7 复合赋值运算符	(34)
2.5.8 逗号运算符	(35)
2.5.9 条件运算符	(36)
2.5.10 指针和地址运算符	(36)
2.5.11 强制类型转换运算符	(37)
2.5.12 sizeof 运算符	(38)
§ 2.6 表达式语句	(39)
§ 2.7 复合语句	(40)
§ 2.8 条件语句	(41)
§ 2.9 开关语句	(43)
§ 2.10 循环语句	(45)
2.10.1 while 语句	(45)
2.10.2 do-while 语句	(46)
2.10.3 for 语句	(47)
2.10.4 goto 语句	(48)
2.10.5 continue 语句	(50)
§ 2.11 返回语句	(50)

第三章 函数	(53)
§ 3.1 函数的定义	(53)
§ 3.2 函数的调用	(55)
3.2.1 函数的调用形式	(55)
3.2.2 对被调用函数的说明	(56)
3.2.3 函数的参数和函数的返回值	(57)
3.2.4 实际参数的传递方式	(59)
§ 3.3 函数的递归调用与再入函数	(60)
§ 3.4 中断服务函数与寄存器组定义	(61)
§ 3.5 变量的存储方式	(65)
3.5.1 局部变量与全局变量	(65)
3.5.2 变量的存储种类	(66)
3.5.3 函数的参数和局部变量的存储器模式	(73)
第四章 数组和指针	(75)
§ 4.1 数组的定义和引用	(75)
§ 4.2 字符数组	(76)
§ 4.3 数组元素赋初值	(77)
§ 4.4 数组作为函数的参数	(79)
§ 4.5 指针的概念	(82)
4.5.1 指针与地址	(82)
4.5.2 指针变量的定义	(83)
4.5.3 指针变量的引用	(85)
4.5.4 指针变量作为函数的参数	(88)
§ 4.6 数组的指针	(89)
4.6.1 用指针引用数组元素	(89)
4.6.2 字符数组指针	(91)
§ 4.7 指针的地址计算	(92)
§ 4.8 函数型指针	(94)
§ 4.9 返回指针型数据的函数	(96)
§ 4.10 指针数组与指针型指针	(97)
4.10.1 指针数组	(97)
4.10.2 指针型指针	(101)
§ 4.11 抽象型指针	(103)
第五章 结构与联合	(107)
§ 5.1 结构的概念	(107)
5.1.1 结构变量的定义	(107)
5.1.2 结构变量的引用	(109)

5.1.3	结构变量的初值	(110)
5.1.4	结构数组	(111)
§ 5.2	结构型指针	(112)
5.2.1	结构型指针的概念	(112)
5.2.2	用结构型指针引用结构元素	(112)
§ 5.3	结构与函数	(113)
5.3.1	将结构作为函数的参数	(113)
5.3.2	将结构型指针作为函数的参数	(116)
§ 5.4	联合	(117)
5.4.1	联合的定义	(117)
5.4.2	联合变量的引用	(118)
第六章	预处理器	(123)
§ 6.1	宏定义	(123)
6.1.1	不带参数的宏定义	(123)
6.1.2	带参数的宏定义	(125)
§ 6.2	文件包含	(127)
§ 6.3	条件编译	(128)
§ 6.4	其他预处理命令	(130)
第七章	C51 编译器	(133)
§ 7.1	C51 编译器控制指令详解	(134)
7.1.1	源控制指令	(135)
7.1.2	列表控制指令	(137)
7.1.3	目标控制指令	(143)
§ 7.2	C51 编译器的数据调用协议	(150)
7.2.1	数据在内存中的存储格式	(150)
7.2.2	目标代码的段管理	(151)
7.2.3	再入函数的栈结构	(153)
7.2.4	与汇编语言程序的接口	(153)
7.2.5	与 PL/M51 语言程序的接口	(167)
7.2.6	配置文件	(167)
7.2.7	C51 编译器的限制	(174)
§ 7.3	C51 编译器的出错信息	(175)
7.3.1	致命错误	(175)
7.3.2	语法及语义错误	(177)
§ 7.4	C51 的库函数	(187)
7.4.1	字符函数 CTYPE.H	(188)
7.4.2	一般 I/O 函数 STDIO.H	(190)
7.4.3	字符串函数 STRING.H	(193)

7.4.4	标准函数 STDLIB.H	(196)
7.4.5	数学函数 MATH.H	(197)
7.4.6	绝对地址访问 ABSACC.H	(199)
7.4.7	内部函数 INTRINS.H	(200)
7.4.8	变量参数表 STDARG.H	(201)
7.4.9	全程跳转 SETJMP.H	(202)
7.4.10	访问 SFR 和 SFR_bit 地址 REGxxx.H	(203)
§ 7.5	C51 编译器的特殊扩充	(203)
7.5.1	对西门子 80C517/537 的特殊扩充	(203)
7.5.2	对 AMD 8xC51 的特殊扩充	(210)
7.5.3	对 PHILIPS 8xC751 的特殊扩充	(210)
第八章	目标文件的连接与转换	(213)
§ 8.1	连接定位器 L51	(213)
8.1.1	连接/定位控制指令	(215)
8.1.2	L51 的出错信息	(239)
§ 8.2	符号转换程序 OHS51	(246)
8.2.1	Intel HEX 文件的格式	(246)
8.2.2	OHS51 的控制指令与出错信息	(247)
§ 8.3	库管理程序 LIB51	(248)
§ 8.4	分组连接定位器 BL51	(251)
8.4.1	BL51 的控制指令	(252)
8.4.2	BL51 的分组配置	(254)
8.4.3	BL51 的出错信息	(282)
8.4.4	分组目标文件转换程序 OC51	(282)
§ 8.5	宏汇编器 A51	(283)
8.5.1	符号与表达式	(286)
8.5.2	汇编伪指令	(288)
8.5.3	宏处理	(294)
8.5.4	汇编控制指令	(301)
8.5.5	A51 的出错信息	(308)
第九章	DS51 软件仿真器	(315)
§ 9.1	DS51 的启动和屏幕窗口分配	(315)
§ 9.2	命令解释软件	(319)
§ 9.3	鼠标器的使用	(320)
9.3.1	用鼠标器输入命令	(320)
9.3.2	用鼠标器执行命令	(320)
§ 9.4	下拉菜单的操作	(321)
9.4.1	选项菜单(Alt+O)	(321)

9.4.2	键盘菜单(Alt+K)	(322)
9.4.3	观察菜单(Alt+V)	(322)
9.4.4	外围集成功能菜单(Alt+P)	(323)
9.4.5	映象菜单(Alt+M)	(324)
9.4.6	帮助菜单(Alt+H)	(324)
9.4.7	CPU 菜单(Alt+C)	(324)
§ 9.5	DS51 的命令	(325)
9.5.1	显示和更新存储器内容命令	(325)
9.5.2	用户程序执行控制命令	(329)
9.5.3	断点管理命令	(329)
9.5.4	其它通用命令	(332)
§ 9.6	表达式	(338)
9.6.1	表达式的组成	(338)
9.6.2	DS51 表达式与 C 表达式之间的差别	(341)
9.6.3	DS51 表达式应用示例	(342)
§ 9.7	DS51 的函数	(346)
9.7.1	用户函数	(346)
9.7.2	内部函数	(348)
9.7.3	信号函数	(351)
9.7.4	DS51 函数与 C 语言函数的差别	(353)
§ 9.8	DS51 的出错信息	(353)
§ 9.9	DS51 的 IOF 驱动软件	(359)

第十章 C51 应用编程技巧与实例 (365)

§ 10.1	C51 应用程序设计的基本方法	(365)
§ 10.2	C51 应用中的一些常见问题与解决方法	(366)
10.2.1	C 语言程序设计中容易出错的地方	(366)
10.2.2	有关 C51 的若干实际应用技巧	(368)
§ 10.3	8051 单片机串行接口扩展应用编程	(374)
10.3.1	用 8051 串行口扩展的矩阵键盘	(374)
10.3.2	利用 8051 串行口实现多机通信	(376)
10.3.3	串行接口五位 LED 驱动器 MC14489 的应用	(379)
§ 10.4	软件模拟 I ² C 总线的 C51 读写程序	(383)
10.4.1	I ² C 总线简介	(383)
10.4.2	I ² C 总线通用读写程序	(386)
§ 10.5	8051 单片机并行接口扩展应用编程	(393)
10.5.1	打印输出接口及其驱动程序	(393)
10.5.2	D/A 及 A/D 转换接口及其驱动程序	(396)
10.5.3	用可编程芯片 8155 实现 I/O 接口扩展	(400)
§ 10.6	80C552 单片机中 A/D 转换器的应用	(416)

10.6.1	80C552 单片机简介	(416)
10.6.2	80C552 内部 ADC 的应用.....	(419)
10.6.3	使用 80C552 内部 ADC 时印刷电路版的设计要点	(423)
10.6.4	使用 80C552 内部 ADC 的 C51 驱动程序	(424)
§ 10.7	87C752 单片机在气流量测量中的应用	(430)
10.7.1	87C752 单片机简介	(430)
10.7.2	气流量测量仪表的硬件设计.....	(435)
10.7.3	气流量测量仪表的软件设计.....	(438)
参考文献.....		(450)

第一章 C 语言的基本知识

§ 1.1 C 语言的特点及其程序结构

C 语言是一种通用的计算机程序设计语言，在国际上十分流行，它既用来编写计算机的系统程序，也可用来编写一般的应用程序。以前计算机的系统软件主要是用汇编语言编写的，对于单片机应用系统来说更是如此。由于汇编语言程序的可读性和可移植都较差，采用汇编语言编写单片机应用系统程序的周期长，而且调试和排错也比较困难。为了提高编制计算机系统和应用程序的效率，改善程序的可读性和可移植性，最好采用高级语言编程。一般的高级语言难以实现汇编语言对于计算机硬件直接进行操作（如对内存地址的操作、移位操作等）的功能。而 C 语言既具有一般高级语言的特点，又能直接对计算机的硬件进行操作，并且采用 C 语言编写的程序能够很容易地在不同类型的计算机之间进行移植，因此 C 语言的应用范围越来越广泛。

与其它计算机高级语言相比，C 语言具有它自身的特点。可以用 C 语言来编写科学计算或其它应用程序，但 C 语言更适合于编写计算机的操作系统程序以及其它一些需要对机器硬件进行操作的场合，有的大型应用软件也采用 C 语言进行编写。这主要是因为 C 语言具有很好的可移植性和硬件控制能力，表达和运算能力也较强。许多以前只能采用汇编语言来解决的问题现在可以改用 C 语言来解决。概括起来说，C 语言具有以下一些特点：

1. 语言简洁，使用方便灵活。C 语言是现有程序设计语言中规模最小的语言之一，而小的语言体系往往能设计出较好的程序。C 语言的关键字很少，ANSI C 标准一共只有 32 个关键字，9 种控制语句，压缩了一切不必要的成分。C 语言的书写形式比较自由，表示方法简洁，使用一些简单的方法就可以构造出相当复杂的数据类型和程序结构。

2. 可移植性好。用过汇编语言的读者都知道，即使是功能完全相同的一种程序，对于不同的机器，必须采用不同的汇编语言来编写。这是因为汇编语言完全依赖于机器硬件，因而具有不可移植性的原因。C 语言是通过编译来得到可执行代码的，统计资料表明，不同机器上的 C 语言编译程序 80% 的代码是公共的，C 语言的编译程序便于移植，从而使在一种机器上使用的 C 语言程序，可以不加修改或稍加修改即可方便地移植到另一种机器上去。

3. 表达能力强。C 语言具有丰富的数据结构类型和多种运算符，可以根据需要采用整型、实型、字符型、数组类型、指针类型、结构类型、联合类型等多种数据类型来实现各种复杂数据结构的运算。C 语言还具有多种运算符，灵活使用各种运算符可以实现其它高级语言难以实现的运算。

4. 表达方式灵活。利用 C 语言提供的多种运算符，可以组成各种表达式，还可采用多种方法来获得表达式的值，从而使用户在程序设计中具有更大的灵活性。C 语言的语法规则不太严格，程序设计的自由度比较大，程序的书写格式自由灵活。程序主要用小写字母来编写，而小写字母是比较容易阅读的，这些充分体现了 C 语言灵活、方便和实用的特点。

5. 可进行结构化程序设计。C 语言是以函数作为程序设计的基本单位的，C 语言程序中的函数相当于一般语言中的子程序。C 语言对于输入和输出的处理也是通过函数调用来实现的。各种 C 语言编译器都会提供一个函数库，其中包含有许多标准函数，如各种数学函数、标准输入输出函数等。此外 C 语言还具有自定义函数的功能，用户可以根据自己的需要编制满足某种特殊需要的自定义函数。实际上 C 语言程序就是由许多个函数组成的，一个函数即相当于一个程序模块，因此 C 语言可以很容易地进行结构化程序设计。

6. 可以直接操作计算机硬件。C 语言具有直接访问机器物理地址的能力，美国 Franklin 公司的 C51 编译器可以直接对 8051 单片机的内部特殊功能寄存器和 I/O 口进行操作，可以直接访问片内或片外存储器，还可以进行各种位操作。

7. 生成的目标代码质量高。众所周知，汇编语言程序目标代码的效率是最高的，这就是为什么汇编语言仍是编写计算机系统软件的主要工具的原因。但是统计表明，对于同一个问题，用 C 语言编写的程序生成代码的效率仅比用汇编语言编写的程序低 10~20%，美国 Franklin 软件公司的 C51 编译器，能够产生及其简洁形式、效率极高的程序代码，在代码质量上可以与汇编语言程序相媲美。

尽管 C 语言具有很多的优点，但和其它任何一种程序设计语言一样也有其自身的缺点，如不能自动检查数组的边界，各种运算符的优先级别太多，某些运算符具有多种用途等。但总的来说，C 语言的优点远远超过了它的缺点，经验表明，程序设计人员一旦学会使用 C 语言之后，就会对它爱不释手，尤其是单片机应用系统的程序设计人员更是如此。

C 语言程序是由若干个函数单元组成的，每个函数都是完成某个特殊任务的子程序段。组成一个程序的若干个函数可以保存在一个源程序文件中，也可以保存在几个源程序文件中，最后再将它们连接在一起。C 语言源程序文件的扩展名为“.c”，如 EX1-1.c，EX1-2.c 等。

一个 C 语言程序必须有而且只能有一个名为 main() 的函数，它是一个特殊的函数，也称为该程序的主函数，程序的执行都是从 main() 函数开始的。下面我们先来看一个简单的程序例子。

例 1.1 已知 $x=10$ ， $y=20$ ，计算 $z=x+y$ 的结果。

```
main()           /* 主函数名 */
{
    int x, y, z;   /* 主函数的内部变量类型说明 */
    x=10; y=20;    /* 变量赋值 */
    z=x+y;         /* 计算 z=x+y 的值 */
}                 /* 程序结束 */
```

本例中 main 是主函数名，要执行的主函数的内容称为主函数体，主函数体用花括弧号“{ }”围起来。函数体中包含若干条将被执行的程序语句，每条语句都必须以分号“;”为结束符。为了使程序便于阅读和理解，可以给程序加上一些注释。C 语言的注释部分由符号“/*”开始，由符号“*/”结束，在“/*”和“*/”之间的内容即为注释内容，注释内容可在一行写完，也可以分成几行来写。注释部分不参加编译，编译时注释的内容不产生可执行代码。注释在程序中的作用是很重要的，一个好的程序设计者应该在程序中使用足够的注释来说明整个程序的功能、有关算法和注意事项等。需要注意的是，C 语言中的注释不能嵌套，即在“/*”和“*/”之间不允许再次出现“/*”和“*/”。

本例的程序是很简单的，它只有一个主函数 main()。在更一般的情况下，一个 C 语言程序除了必须有一个主函数之外，还可能有若干个其它的功能函数。下面我们再来看一个例子。

例 1.2 求最大值。

```
#include<stdio.h>           /* 预处理命令 */
#include<reg51.h>
main()                       /* 主函数名 */
{                             /* 主函数体开始 */
    int a, A, c;              /* 主函数的内部变量类型说明 */
    int max(int x,int y);      /* 功能函数 max 及其形式参数说明 */
    SCON=0x52;                /* 8051 单片机串行口初始化 */
    TMOD=0x20;
    TCON=0x69;
    TH1=0x0F3;
    scanf("%d %d", &a, &A);   /* 输入变量 a 和 b 的值 */
    c=max(a,A);               /* 调用 max 函数 */
    printf("max=%d", c);      /* 输出变量 c 的值 */
}                             /* 主程序结束 */
int max(int x,int y)         /* 定义 max 函数，x、y 为形式参数 */
{                             /* max 函数体开始 */
    int z;                    /* max 函数内部变量类型说明 */
    if ( x>y ) z=x;           /* 计算最大值 */
    else z=y;
    return(z);                /* 将计算得到的最大值 z 返回到调用处 */
}                             /* max 函数结束 */
```

在本例程序的开始处使用了预处理命令 #include，它告诉编译器在编译时将头文件 stdio.h 和 reg51.h 读入后一起编译。在头文件 stdio.h 中包括了对标准输入输出函数的说明，在头文件 reg51.h 中包括了对 8051 单片机特殊功能寄存器的说明。本程序中除了 main() 函数之外，还用到了功能函数调用。函数 max 是一个被调用的功能函数，其作用是将变量 x 和 y 中较大者的值赋给变量 z，并通过 return 语句将它的值返回到 main() 函数的调用处。变量 x 和 y 在函数 max 中是一种形式变量，它的实际值是通过 main() 函数中的调用语句传送过来的。此外，ANSI C 标准规定函数必须要“先说明，后调用”，因此在 main() 函数的开始处，将函数 max 与变量一起进行了说明。函数是 C 语言程序的基本单位，函数调用类似于子程序调用，用户可以根据实际需要编制出各种不同用途的功能函数。C 语言编译器还提供了十分丰富的库函数，可以说 C 语言是函数式的语言，利用 C 语言的这一特点，可以很容易实现结构化程序设计。

本例在 main() 函数中调用了库函数 scanf 和 printf，它们分别是输入库函数和输出库函数。C 语言本身没有输入输出功能，输入输出是通过函数调用来实现的。需要说明一点的是，Franklin C51 提供的输入输出库函数是通过 8051 系列单片机的串行口来实现输入输出的，因此在调用库函数 scanf 和 printf 之前，必须先对 8051 单片机的串行口进行初始化。另外我们在程序中还可以看到小写字母 a 和大写字母 A，它们分别是两种不同的变量，C 语言规定同一个字母由于其大小写的不同可以代表两个不同的变量，这也是 C 语言的一个特点。一般的习惯是在普通情况下采用小写字母，对于一些具有特殊意义的变量或常数采用大写字母，如本

例中所用到的 8051 单片机特殊功能寄存器 SCON、TMOD、TCON 和 TH1 等。但是必须注意的是在 C 语言程序中同一字母的大小写是有区别的，例如 SCON 和 scon 在 C 语言程序中会被认为是两个完全不同的变量。

从以上两个例子可以看到，一般 C 语言程序具有如下的结构：

预处理命令	#include< >
函数说明	long fun1(); float fun2();
功能函数 1	fun1() { 函数体 ... }
主函数	main() { 主函数体 ... }
功能函数 2	fun2() { 函数体 ... } . . .

C 语言程序的开始部分通常是预处理命令，如上面程序中的 #include 命令。这个预处理命令通知编译器在对程序进行编译时，将所需要的头文件读入后再一起进行编译。一般在“头文件”中包含有程序在编译时的一些必要的信息，通常 C 语言编译器都会提供若干个不同用途的头文件。头文件的读入是在对程序进行编译时才完成的。此外还有其它一些预处理命令，将在第六章中详细介绍。

C 语言程序是由函数所组成的。一个 C 语言程序至少应包含一个主函数 main()，也可以包含一个 main() 函数和若干个其它的功能函数。函数之间可以相互调用，但 main() 函数只能调用其它的功能函数，而不能被其它函数所调用。功能函数可以是 C 语言编译器提供的库函数，也可以由用户按实际需要自行编写。不管 main() 函数处于程序中的什么位置，程序总是从 main() 函数开始执行。

一个函数由“函数定义”和“函数体”两个部分组成。函数定义部分包括有函数类型、函数名、形式参数说明等，函数名后面必须跟一个圆括弧()，形式参数说明在()内进行。函数也可以没有形式参数，如 main()。函数体由一对花括弧“{ }”组成，在“{ }”里面的内容就是函数体，如果一个函数有多个“{ }”，则最外面的一对“{ }”为函数体的范围。函数体的内容为若干条语句，一般有两类语句，一类为说明语句，用来对函数中将要用到的变量进行定义；另一类为执行语句，用来完成一定的功能或算法处理。有的函数体仅有一对“{ }”，其中既没有变量定义语句，也没有执行语句，这也是合法的，称为“空函数”。本书第三章将对函数进行详细讨论。

C 语言源程序可以采用任何一种编辑器来编写，如 EDLIN 或 PE 等。C 语言程序的书写格式十分自由。一条语句可以写成一 行，也可以写成几行；还可以在一行内写多条语句；但是需要注意的是，每条语句都必须以分号“；”作为结束符。另外，C 语言是对大小写字母敏感的，C 语言编译器在对程序进行编译时，对于程序中同一个字母的大小写作为不同的变量来处理。虽然 C 语言程序不要求具有固定的格式，但我们在实际编写程序时还是应该遵守一定的规则，一般应按程序的功能以“缩格”形式来写程序，同时还应在适当的地方加上必要的注释。注释对于比较大的程序来说是十分重要的，一个较大的程序如果没有注释，在过了一段时间之后恐怕连程序编制者自己也难以明白原来程序的内容，更不用说让别人来阅读或修改程序了。

C 语言本身没有输入输出功能，C 语言程序的输入和输出是通过函数调用来实现的。Franklin C51 编译器提供的输入输出库函数 scanf 和 printf 是通过 8051 单片机的串行口实现的，在程序中使用这种输入输出库函数之前必须先对 8051 单片机的串行口进行初始化。但是对于单片机应用系统来说，由于具体要求的不同，应用系统的输入输出方式多种多样，不可能一律采用串行口作输入和输出。因此应该根据实际需要，由应用系统的研制人员自己来编写满足特定需要的输入输出函数，这一点对于单片机应用系统的开发研制人员来说是十分重要的。

§ 1.2 C 语言的标识符和关键字

C 语言的标识符是用来标识源程序中某个对象名字的。这些对象可以是函数、变量、常量、数组、数据类型、存储方式、语句等。一个标识符由字符串、数字和下划线等组成，第一个字符必须是字母或下划线，通常以下划线开头的标识符是编译系统专用的，因此在编写 C 语言源程序时一般不要使用以下划线开头的标识符，而将下划线用作分段符。C51 编译器规定标识符最长可达 255 个字符，但只有前面 32 个字符在编译时有效，因此在编写源程序时标识符的长度不要超过 32 个字符，这对于一般应用程序来说已经足够了。前面已经指出，C 语言是对大小写字母敏感的，如“max”与“MAX”是两个完全不同的标识符。程序中对于标识符的命名应当简洁明了，含义清晰，便于阅读理解，如用标识符“max”表示最大值，用“TIMER0”表示定时器 0 等。

关键字是一类具有固定名称和特定含义的特殊标识符，有时又称为保留字。在编写 C 语言源程序时一般不允许将关键字另作别用，换句话说就是对于标识符的命名不要与关键字相同。与其它计算机语言相比，C 语言的关键字是比较少的，ANSI C 标准一共规定了 32 个关键字，表 1.1 按用途列出了 ANSI C 标准的关键字。

表 1.1 ANSI C 标准的关键字

关 键 字	用 途	说 明
auto	存储种类说明	用以说明局部变量，缺省值为此
break	程序语句	退出最内层循环体
case	程序语句	switch 语句中的选择项
char	数据类型说明	单字节整型数或字符型数据
const	存储类型说明	在程序执行过程中不可修改的变量值

续表

关键字	用途	说明
continue	程序语句	转向下一次循环
default	程序语句	switch 语句中的失败选择项
do	程序语句	构成 do...while 循环结构
double	数据类型说明	双精度浮点数
else	程序语句	构成 if...else 选择结构
enum	数据类型说明	枚举
extern	存储种类说明	在其它程序模块中说明了的全局变量
float	数据类型说明	单精度浮点数
for	程序语句	构成 for 循环结构
goto	程序语句	构成 goto 转移结构
if	程序语句	构成 if...else 选择结构
int	数据类型说明	基本整型数
long	数据类型说明	长整型数
register	存储种类说明	使用 CPU 内部寄存器的变量
return	程序语句	函数返回
short	数据类型说明	短整型数
signed	数据类型说明	有符号数, 二进制数据的最高位为符号位
sizeof	运算符	计算表达式或数据类型的字节数
static	存储种类说明	静态变量
struct	数据类型说明	结构类型数据
switch	程序语句	构成 switch 选择结构
typedef	数据类型说明	重新进行数据类型定义
union	数据类型说明	联合类型数据
unsigned	数据类型说明	无符号数据
void	数据类型说明	无类型数据
volatile	数据类型说明	说明该变量在程序执行中可被隐含地改变
while	程序语句	构成 while 和 do...while 循环结构

C51 编译器除了支持 ANSI C 标准的关键字以外, 还扩展了如表 1.2 所示的关键字:

表 1.2 C51 编译器的扩展关键字

关键字	用途	说明
bit	位标量声明	声明一个位标量或位类型的函数
sbit	位变量声明	声明一个可位寻址变量
sfr	特殊功能寄存器声明	声明一个特殊功能寄存器(8 位)
sfr16	特殊功能寄存器声明	声明一个 16 位的特殊功能寄存器
data	存储器类型说明	直接寻址的 8051 内部数据存储器
bdata	存储器类型说明	可位寻址的 8051 内部数据存储器